http://www.moderntechno.de/index.php/meit/article/view/meit35-00-042
DOI: 10.30890/2567-5273.2024-35-00-042

AI IN THE CONSTRUCTION OF EDUCATIONAL TOOLS AND QUIZZES FOR DIFFERENT PROGRAMMING LANGUAGES

Korzh Mykhailo
2nd-year master's degree student, Software Engineering
ORCID: 0009-0009-9387-8632
Tytenko Sergiy
Ph.D., Associate Professor
ORCID: 0000-0002-7548-9053
American University Kyiv,
Ukraine, Kyiv, Poshtova Pl. 3, 02000

Abstract. *With the development of AI, especially large language models (LLM), education has undergone significant changes. This article explores the role of AI-based tools in generating educational quizzes and learning materials for programming languages such as Python, Java, and C++. LLM combined with natural language processing (NLP) enables personalized learning experiences by generating adaptive tests and providing real-time feedback. The article discusses the benefits, challenges, and future directions of AI in programming education, highlighting the potential of advanced scalable learning tools.*

Keywords: *AI, large language model, quiz generation, personalized learning, programming languages, natural language processing, Python, Java, C++, adaptive learning, educational tools.*

Introduction.

Artificial intelligence (AI) will become a cornerstone of education, significantly changing the way students and teachers interact with learning materials and each other. One of AI's most potentially powerful applications is the creation of educational tools and quizzes specifically designed for different programming languages. These tools will improve learning outcomes by providing adaptive, personalized, and engaging content that meets the needs of each student [1,2].

Programming languages are often complex and require a deep understanding of both theoretical concepts and practical applications. To help students master these skills, AI-powered educational tools use large language models (LLM) and natural language processing (NLP) to create quizzes and exercises. These tools not only automate the labor-intensive process of quizzes creation but also provide real-time feedback, constantly adapting to the student's learning style and progress. This article examines the development and application of AI-based educational tools in programming education, focusing on quizzes creation, personalized learning, and the challenges and opportunities of AI in this area.

The Role of AI in Programming Education

Teaching programming constantly poses a variety of and, most importantly, unique challenges associated with the huge variety of languages with different levels of both overall complexity and approaches to working with them. While AI hasn't revolutionized education yet, it has provided many opportunities to automate routine processes in various aspects of education. These advances will allow teachers to spend less time on routine work and more time on individualized student approaches.

Tools that use large language models (LLMs) such as GPT-3, Gemini, and GPT-4o can process learning materials such as presentations, books, lecture notes, and other educational materials. This is particularly useful in both preparing a summary of information and generating questions and test tasks based on the information provided. Natural Language Processing (NLP) plays a key role in the formation of these materials in modern LLMs. It is thanks to high-quality NLP with the ability to understand the context and take into account previous material that a modern LLM can provide high-quality and meaningful results. In addition, automatic quiz generation saves time, and the quizzes cover the lecture material well [3]. This is especially useful in programming, where constant practice and testing of knowledge are essential for developing working skills.

The ability to generate multiple-choice questions has the potential to be a key factor in making programming easier to learn. This can be achieved by being able to tailor questions to different levels of difficulty and specific requirements.

For example, in one study, ChatGPT demonstrated a “moderate positive correlation between the difficulty ratings assigned by ChatGPT-4 and the perceived difficulty ratings given by participants” indicating its ability to generate relevant questions for learners at different levels [4].

AI-Driven Quiz Generation

Using cues, the LLM (ChatGPT) can generate high-quality distractors, with research showing that 58.8% of the distractors generated were rated as high-quality [5]. This ensures that quizzes do not mislead students with meaningless or obviously incorrect options.

Modern multimodal models have opened new possibilities for automating the creation of quizzes for educational purposes [3,6], which allows for a significant reduction in teachers’ time. Also, as highlighted in several studies [3,4], Natural language processing (NLP)-based models can generate questions with different levels of difficulty, making them suitable for different learners with different levels of proficiency [6]. The ability of these models to understand and analyze text at a sufficient level provides a high degree of confidence that the questions generated will be relevant and have educational value. For example, a web application developed in the project "ePub-to-Quiz Conversion with Large Language Models"[6] allows users to upload EPUB documents and create interactive quizzes. This tool clearly demonstrates how AI can be integrated into educational institutions at the existing level of technology.

Another major benefit of NLP-based LLM in educational tools is its potential to personalize learning. By using NLP to process responses in real-time, LLMs can provide rapid feedback (in the form of hints, recommendations for review or further study), thereby improving comprehension. As noted in the study [2] LLMs such as GPT-4 have demonstrated satisfactory results in generating accurate multiple-choice quizzes.

Recent research [1,4,6,7] shows that by tuning the model and properly organizing the prompt, LLM can be configured to create complex and effective quizzes.

Potentially, quizzes can be adjusted in real time based on previous answers, thereby focusing on areas of knowledge where the student needs to deepen their knowledge. This approach to quizzes generation can improve and reduce the speed and quality of learning.

Challenges and Limitations of AI in Programming Education

Despite the many benefits of AI-based educational tools, there are a few challenges and limitations when implementing them in programming education:

Quality control: While AI can automate quiz generation, there is always a risk of generating questions with errors or ambiguities. AI-generated quizzes require human review to ensure that the questions are technically accurate and relevant to the learner's goals [3]. Additionally, because programming often requires a sophisticated understanding of algorithms and data structures, it can be difficult for AI systems to create exercises that are both challenging and educationally useful [4].

Bias in AI models: AI models are trained on large data sets, and these data sets can carry inherent biases [8]. In the context of learning to code, this can lead to questions or explanations that favor certain languages, practices, or paradigms over others. This can limit the diversity of learning materials presented to students and potentially skew their understanding of programming best practices [3].

Technical limitations: AI-based educational tools, especially those that rely on LLM, are computationally expensive. Running these models can be expensive, especially in real-time applications where immediate feedback is needed. Additionally, while some models, like GPT-4, have extensive training, they still have limitations in terms of the amount of content they can process at one time, which can prevent them from creating comprehensive quizzes from long learning materials [3, 9].

Over-reliance on AI: There is a risk that students may become overly reliant on AI-generated feedback and tests, potentially reducing the role of critical thinking and self-assessment in the learning process. While AI tools provide valuable support, educators must ensure that students also develop the ability to critically assess their own work, rather than always relying on automated systems [3].

Lack of Reflective Judgment: A major problem with AIs is their limited ability to identify incorrect or irrelevant answer options. Many LLMs, such as GPT-4, tend to blindly follow instructions and may attempt to select an answer even if none of the options are correct. This limitation, known as a lack of "reflective judgment," limits the model's ability to handle complex questions. This problem is especially relevant in learning to code, where accurate interpretation and critical analysis are critical to both creating quizzes questions and assessing their correctness [10].

Future Directions for AI in Programming Education

As AI technology continues to advance, we can expect to see several interesting future applications in programming education:

Advanced Adaptive Learning: The next generation of AI-powered educational tools will benefit from further advances in adaptive learning algorithms. These systems should be able to develop a deeper, more nuanced understanding of individual learners' needs, allowing them to adjust the difficulty and focus of tests in real-time. With more advanced personalization, AI tools will be able to provide fully customizable learning paths that adapt not only to a learner's skill level but also to their preferred learning style, ensuring maximum engagement and effectiveness.

Support for more niche programming languages: While AI-powered tools currently focus on popular programming languages like Python, Java, and C++ [11], future AI models will need to expand their capabilities to more specialized languages such as Rust, Haskell, and Go. This will provide students with more diverse learning opportunities and better preparation for different programming paradigms and technologies.

***Integration with hands-on coding platforms*:** Future AI-powered educational tools will need to integrate more seamlessly with interactive coding platforms like Codecademy. This will allow students to practice coding exercises directly within the AI system, receiving immediate feedback on code execution, efficiency, and style. By combining theoretical quizzes with hands-on coding challenges, AI tools will offer a more holistic approach to coding education.

***Improved feedback mechanisms*:** As AI models become more sophisticated, they will need to provide more detailed and meaningful feedback. Instead of simply marking an answer as correct or incorrect, AI tools should be able to deeply analyze a student's code, offering feedback on performance optimization, code readability, and adherence to best practices. This type of feedback will be invaluable to students, especially those who aspire to become professional developers.

***AI for Collaborative Learning*:** Future AI-powered tools could facilitate collaborative learning by allowing students to work on AI-assisted group projects. By analyzing group dynamics and individual contributions, AI could help identify areas where certain students need more support or where collaboration strategies could be improved. This would be especially useful in team-based programming environments, where effective collaboration is key to success.

***Enabling multimodal learning*:** Future AI tools should support multimodal learning methods that combine text, audio, and visual learning materials. For example, AI should be able to generate quizzes based on video lessons, analyze students' code execution during real-time coding sessions, or provide real-time visualization of code performance.

Summary and Conclusions.

The integration of AI into programming education, especially through large language models (LLM) and natural language processing (NLP), is revolutionizing the way students and teachers interact with learning materials. This article explores how AI-powered tools automate quiz generation, provide personalized learning experiences, and adapt to individual learners' needs. Despite challenges such as contextual accuracy and technical limitations, AI brings undeniable benefits including efficiency, scalability, and personalized feedback. Relevant areas for further research include personalized student testing, the generation of quiz tasks in different programming languages, and their further integration into existing educational platforms. An important area of further research should be the construction of a reliable architecture for such solutions with an emphasis on ease of use and a high level of data protection.

References:

1. Hasan, A. S. M. M. E., Shahnoor, M. A., Tasneem, K. B., & Sumaiya, S. (2024). Automatic question & answer generation using generative Large Language Model (LLM). BRAC University Institutional Repository. URL:

https://dspace.bracu.ac.bd:8443/xmlui/handle/10361/22833

2. Sreekanth, D. (2023). AI-based quiz generation: The role of LLMs in digital education. Kennesaw State University Digital Commons. URL:https://digitalcommons.kennesaw.edu/cday/2023fall/Undergraduate_Research/7/

3. Meißner, Niklas; Speth, Sandro; Kieslinger, Julian; Becker, Steffen (2024): EvalQuiz – LLM-based Automated Generation of Self-Assessment Quizzes in Software Engineering Education. Software Engineering im Unterricht der Hochschulen 2024. DOI: 10.18420/seuh2024_04. Bonn: Gesellschaft für Informatik e.V.. PISSN: 1617-5468. ISBN: 978-3-88579-255-0. pp. 53-64. Linz, Österreich. 29. Februar - 1.März 2024. URL: https://dl.gi.de/items/1561864d-a5b9-42d2-821d-41baf80eb630/

4. Lopez, C., Morrison, M., & Deacon, M. (2024). Language models for generating programming questions with varying difficulty levels. European Public & Social Innovation Review, 9, 1–19. https://doi.org/10.31637/epsir-2024-760

5. Bitew, S. K., Deleu, J., Develder, C., & Demeester, T. (2023). Distractor generation for multiple-choice questions with predictive prompting and large language models. *arXiv*. URL: https://arxiv.org/abs/2307.16338

6. Ersoy, B. I. (2024). ePub-to-Quiz Conversion with Large Language Models. URL: https://www.cl.uzh.ch/dam/jcr:5c3787db-bc0f-42a6-82ab-92fbc435585e/masterarbeit.final.pdf

7. Elkins, S., Kochmar, E., Cheung, J. C. K., & Serban, I. (2024). How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes. Proceedings of the AAAI Conference on Artificial Intelligence, 38(21), 23084-23091. https://doi.org/10.1609/aaai.v38i21.30353

8. Pagano, T. P., Loureiro, R. B., Lisboa, F. V. N., Peixoto, R. M., Guimarães, G. A. S., Cruz, G. O. R., Araujo, M. M., Santos, L. L., Cruz, M. A. S., Oliveira, E. L. S., Winkler, I., & Nascimento, E. G. S. (2023). Bias and Unfairness in Machine Learning Models: A Systematic Review on Datasets, Tools, Fairness Metrics, and Identification and Mitigation Methods. URL: https://www.mdpi.com/2504-2289/7/1/15

9. Frankford, E., Höhn, I., Sauerwein, C., & Breu, R. (2024). A Survey Study on the State of the Art of Programming Exercise Generation using Large Language Models. *arXiv*. URL: https://arxiv.org/abs/2405.20183

10. Góral, G., Wiśnios, E., Sankowski, P., & Budzianowski, P. (2024).When All Options Are Wrong: Evaluating Large Language Model Robustness with Incorrect Multiple-Choice Options. *arXiv*. URL: https://arxiv.org/abs/2409.00113

11. Diehl, P., Nader, N., Brandt, S., & Kaiser, H. (2024). Evaluating AI-generated code for C++, Fortran, Go, Java, Julia, Matlab, Python, R, and Rust. arXiv. URL:https://arxiv.org/abs/2405.13101